On-chip communication for neuro-glia networks

On-chip Communication for Neuro-Glia Networks

George Martin [1*], Jim Harkin, Liam J. McDaid, John J. Wade, Junxiu Liu

[1] School of Computing, Engineering and Intelligent Systems, Ulster University, Magee, Derry, Northern Ireland
[*] ({martin-g11, jg.harkin, lj.mcdaid, jj.wade, j.liu1} @ulster.ac.uk)

Abstract: Hardware has become more prone to faults as a result of geometric scaling, wear-out and faults caused during the manufacturing process, therefore, the reliability of hardware is reliant on the need to continually adapt to faults. A computational model of biological self-repair in the brain, derived from observing the role of astrocytes (a glial cell found in the mammalian brain), has captured self-repair within models of neural networks known as neuro-glia networks. This astrocyte-driven repair process can address the issues of faulty synapse connections between neurons. These astrocyte cells are distributed throughout a neuro-glia network and regulate synaptic activity, and it has been observed in computational models that this can result in a fine-grained self-repair process. Therefore, mapping neuro-glia networks to hardware provides a strategy for achieving self-repair in hardware. The internal interconnecting of these networks in hardware is a challenge. Previous work has focused on addressing neuron to astrocyte communication (local), however, the global self-repair process is dependent on the communication infrastructure between astrocyte-to-astrocyte; e.g. astrocyte network. This paper addresses the key challenge of providing a scalable communication interconnect for global astrocyte network requirements and how it integrates with existing local communication mechanism. Area/power results demonstrate scalable implementations with the ring topology while meeting timing requirements.

1. Introduction

Components have an increased risk to faults due to geometric scaling or physical defects in the silicon caused during manufacturing [1], [2]. Fault tolerance or self-repair techniques may be applied to reduce the risk of faults affecting a system's operational functionality. This is particularly important in mission critical systems [3]–[6]. Existing approaches suffer from limited granularity as fine-grained approaches incur large area overhead, e.g. Triple Mode Redundancy (TMR). TMR is implemented at gate or component level and suffers from a lack of granularity as only the integral components are protected. Ultimately TMR incurs a large area overhead of critical components (~3 times the overhead per component) [5], [7]–[9]. TMR relies on the use of a comparator to mask faults, discovering and detecting faults is another issue. Current repair strategies use specific hardware to discover faults and subsequently correct them [10]–[13]. A centralized repair mechanism is flawed for a number of reasons 1. It causes an increase in area overhead and system complexity 2. It has a limited scope of repair, which may not include low level faults and 3. If the repair agent suffers a fault then the system is fundamentally compromised. Self-repair is therefore, a desired trait in hardware, with an emphasis on a fine grained and distributed capability. Several self-repair mechanisms have been explored and include online detection/correction and autonomous self-repair, although they all incur large overheads and further the component/system complexity [13]–[15].

The biological process of self-repair within the brain is a function of the astrocyte. The astrocyte has been identified as a vastly distributed cell within networks of neurons mediating the strength of synaptic connections between neurons [16], [17]. Recent research has shown that astrocytes can provide fine grained repair. They provide a distributed repair network, this process is performed in the brain via astrocyte networks [18], [19]. Computational models displaying this repair process have been successfully captured and applied to spiking neural networks (SNNs) [17]. Self-repair has also been demonstrated on hardware platforms using astrocytes [20]. This work focused on individual neurons and their firing activity. When the probability of release (PR) across the synapse drops, the neurons activity will also drop, which in turn causes a knock-on effect at the output of the network. This lack of activity may be caused by a corresponding fault, a failed synapse or a silent neuron. These faults may be overcome and repaired by increasing the PR of synapses connected to remaining healthy neurons; i.e. as the PR increases the neurons activity will return to a pre-fault level. This is evident when faults, including catastrophic failure (80% of faulty synapses), have been introduced into a neural network [21]. An SNN solely consists of neurons connected via synapses in a complex topology. Neuro-glia networks however, become more complex due to the additional connectivity between astrocytes. There are additional components in the networking infrastructure, neighbouring astrocytes and multiple neurons. This a complex communication structure between astrocytes and neurons. While spiking-based communication in SNNs is a binary event, a spike or no-spike, the communication between astrocytes is a continuous process. It occurs over a much longer timescale. The key focus of this work is to provide a multi-level communication infrastructure between astrocytes. Supporting global communication between astrocytes, as well as supporting local communication between the astrocyte and neurons.

There have been promising implementations of astrocyte cells within neuromorphic circuitry [21], [22] and digital hardware devices [20], [23]–[27]. This work extends on the previous work by the authors [28] where a ring topology was used to communicate e-SP from the astrocyte to associated synapses within HNoC. This supports local communication for self-repair. The NoC was implemented for local

communication between the astrocytes and neurons. This work focuses on global astrocyte network communications which is a higher-level communication requirement of astrocyte-neuron networks.

2. Self-repair in biology

Astrocyte communication is viewed as both local and global. Local communication pathways connect the astrocyte directly to neurons. A global information exchange occurs within the astrocyte infrastructure as a standalone process, this interconnect requirement is difficult to implement in hardware. This neuro-glia structure can be viewed as two separate networks, where pathways are in place to support the information exchanges of both neurons and astrocytes. Neurons exchange information with astrocytes and other neurons via spike events, a one or zero, stimulated by neuron activity. Astrocytes communicate using separate signalling pathways. Spike events between pre- and postsynaptic neurons instigate these signalling exchanges between astrocytes. A neuron is made up of dendrites (inputs) and an axon (output). Spikes are the result of accumulation of charge from neighbouring neurons, each spike accumulates a small charge until it breaches the neurons threshold. This causes a spike to emit from the neuron, through the axon and across the synaptic cleft. There is an intracellular chemical reaction which takes place and astrocytes can modulate or mediate these reactions [18]. This results in the increase or decrease of the Probability of Release (PR) in associated synapses. This equilibrium balances the PR on the synapse. There are two feedback processes between the astrocyte and neuron, namely direct and indirect (See Fig. 1.). The direct process is the e-SP (Endocannabinoid-mediated Synaptic Potentiation). This strengthens the PR within the synapse. The in-direct feedback is referred to as DSE (Depolarization-induced Suppression of Excitation) and decreases the PR of associated synapses. When a spike event arrives from the pre-synaptic neuron, there is a release of glutamate into the synaptic cleft. The glutamate then binds to the receptors on the post-synaptic dendrite. Endocannabinoids or 2-AG (2-arachidonyl glycerol) ions are synthesised and subsequently released from the post-synaptic neuron. These 2-AG ions bind to the astrocyte cell membrane which causes oscillations of calcium (Ca2+) to occur in the cytoplasm.

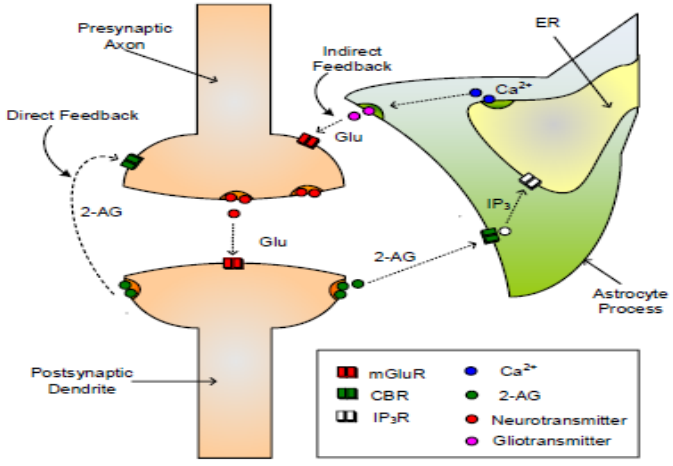

Fig. 1. Direct and indirect feedback across a synaptic cleft during a spike [18].

This self-repair behaviour has been modelled in previous work [18] and is the signalling process which facilitates repair decisions at a low or local level. However, this increases the complexity and signalling processes within the network. Astrocytes are connected via gap junctions or intracellular routes which allow Inositol Triphosphate (IP_3) exchange. The astrocyte network can be viewed as a high-level network, working in parallel with the neural network, responsible for regulating synaptic plasticity through neural networks.

3. Neuro-glia Networks

Computational models of self-repair with astrocytes have been effectively applied to spiking neural networks. Successfully demonstrating self-repair within the neural network. The astrocytes increase the PR of healthy synapses restoring neuron activity [17]. The communication infrastructure allows the astrocyte to regulate the PR of synapses throughout the SNN. When a healthy neuron's firing frequency decreases or it suddenly stops firing, it is considered a faulty neuron. An increase in PR on the remaining healthy synapses can restore functionality to the neuron. This work has shown that astrocyte can detect fine-grained faults (faulty synaptic connections) associated with silent or near-silent neurons [18]. Increasing the weights on the surrounding healthy synapses, enables the healthy synapses to cause a weak neuron (fault-induced weakness) to start firing again. In Fig. 2, two neurons, N1 and N2, are fully operational. A1 is an astrocyte and e-SP and DSE show the signalling pathways. C1 and C2 are the associated synapses for each neuron, typically 10 in total associated with each neuron, N. In (A) N1 and N2 are firing, 2-AG e-SP and DSE levels are maintained. In (B) N2 has stopped its normal firing rate and as a result there are a number of imbalances in the glio-transmitters. DSE from N2 stops, this is the depreciation of the PR at the synapse. The astrocyte maintains the e-SP (the potentiation of the PR at the synapse is still active due to N1 still maintaining its level of activity. Therefore, there is an increase in PR across the healthy synapses. This restores the spiking activity in N2. This self-repair can be observed after catastrophic failure (up to 80% of faults) [17]. This demonstrates how these networks detect and repair faults at fine-grained levels (synapses) via several astrocytes as the repair controller.

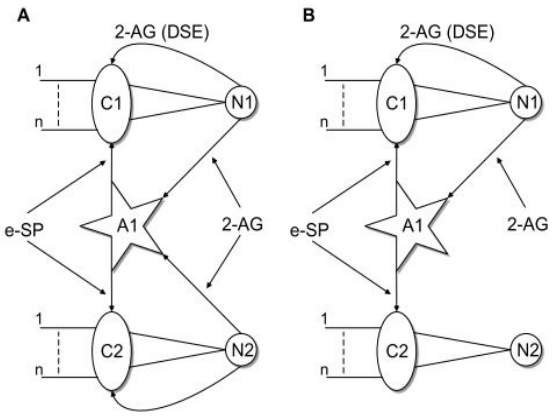

Fig. 2. Direct and indirect feedback maintained by an astrocyte.

Exploring hardware implementations of neuro-glia networks is a natural progression in realising self-repairing systems. The interconnecting of these networks in hardware is a challenge. As a promising solution for a scalable and densely connected hardware platform, Networks-on-Chip (NoCs) have the capability to connect large networks of processing elements (PEs) on a single chip. Using routers and packet-based communication [29]–[32], NoCs are regarded as a suitable interconnect mechanism for SNN hardware [15], [33], [34].

3.1. Hierarchical Networks-on-Chip (HNoC)

The increased number of cores in System on Chip (SoC) and Multi-Processor System on Chip (MPSoC) have increased the complexity of the wiring structure. There is a near-to exponential increase in connections when increasing the number of cores on a single chip. NoCs are based on networking protocols, using a digital interconnect, where information is communicated in the form of packets. The NoC interconnect uses routers to communicate packets of data between cores, where a packet contains the address of the intended processing element/ and the actual application data (payload) [29]–[31]. The NoC is an effective communication protocol developed to reduce interconnect overhead incurred by traditional bus-based systems. HNoC [2] is an interconnect paradigm developed at Ulster University which demonstrates a scalable interconnect solution for hardware SNN implementations. HNoC provides communication for high speed spike events of SNNs. Supporting global and local communication between the astrocyte and neurons. In effect, the aim is to develop communication channels within a neuro-glia network. HNoC consists of three levels of NoC communication and is structured in a hierarchical manner. Using a hybrid of NoC topologies, HNoC exploits the positive qualities of each topology to tailor specifically for different neuron communication levels. For example, level one is the lowest level of HNoC and is the *node facility*. It uses a node router to communicate and connect to a neuron with its immediate neighbours, there are 10 neurons per node facility. It uses a point-to-point or direct NoC topology. This exploits the short communications between neurons and optimises the rate at which neurons can communicate. These node facilities connect to a tile router, 10 nodes per tile, which supports up and downstream communication between neurons within different node facilities, this allows 100 neurons to reside within a tile. HNoC can support the infrastructure of 400 neurons, and communicates with *cluster facilities* to support even larger networks consisting of thousands of neurons in a single network [2]. Replicating the complex structure of neuro-glia networks is difficult in hardware due to vast interconnect requirements. It's challenging, as extra connections are required for astrocyte-to-neurons and astrocyte-to-astrocyte communication. This is in addition to the already complex inter-neuron connectivity requirements. There have been advancements implementing astrocyte cells in neuromorphic systems [21], [22] and digital circuits [23]–[26] with the aim of exploring their behaviour. An astrocyte model recently developed by the authors [27], embeds features of self-repair with neural networks and applies them to hardware applications; e.g. a robotic car controlled by a neural network [20].

The key advantage of NoC for neural hardware is scalable interconnect, e.g. reducing area overhead and lower power with reduced wiring complexity. In SNNs alone, there are high levels of data parallelism between neurons within a network. The SNN has neurons (processing elements), synapses (links) and a complex neuron topology [35]. Fig. 3 outlines a SNN in the format of the HNoC architecture [2]. HNoC is constructed with 10 neurons connected in a single node facility using a star or point-to-point fashion. Each tile facility connects 10 node routers in a tile facility and finally, each cluster facility contains four tile facilities. Therefore, each cluster facility connects 400 neurons. The routing facilities provide HNoC with access to neural facilities in other cluster facilities, thereby allowing a higher number of neurons to be implemented in a neural network. The cluster facility allows connection to other clusters using cardinal points, i.e. North (N), East (E), South (S) and West (W). This allows a scalable neural network to be implemented within HNoC. Using this hierarchical topology, it reduces wiring layouts leading to complex and inefficient routing structures in hardware.

3.2. Networks-on–Chip for Neuro-Glia Hardware

The HNoC hierarchical approach assists in identifying a communication network for astrocytes. Astrocytes communicate on a local and global scale, using routers to distinguish between these local and global communications, they can be separated and viewed in a hierarchical manner. This not only supports parallelism, allowing data to be passed from neuron to neuron and astrocyte to astrocyte, but also the interactions between neurons and astrocytes.

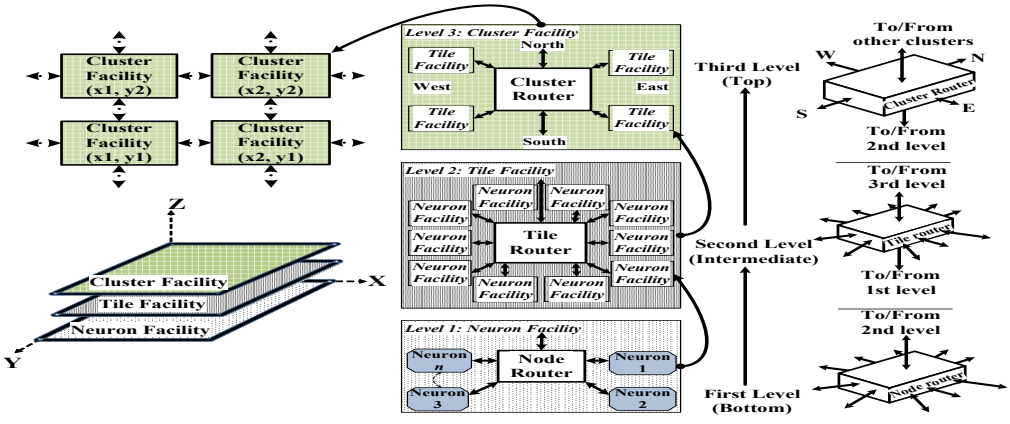

Fig. 3. Detailed diagram displaying the levels of communication within HNoC [2].

Neurons communicate through spike events however, astrocytes communicate with each other through the exchange of IP_3. These are very different communication patterns and speeds which must be adhered to when realizing a neuro glia network. Inspiration from using different topologies such as ring establishes the motivation for exploring the trade-off between reduced communication bandwidth and reduce area overhead from larger buses/wiring.

4. Multi-level Neuro-glia Network Interconnect

Previous work [29] focused on the local communication of e-SP to the neurons in each node facility connected in HNoC. Using a ring topology in the lowest level of the NoC, it provided a balance between achieving a lower area overhead and relaxing the packet latency requirement. Increasing packet latency was tolerable due to the slow 'biologically' rate of communication of astrocyte to synapses/neurons (i.e. order of seconds). The ring topology allows astrocytes to communicate e-Sp with neurons using minimal area, this low level interconnect can be viewed as the local communication. Fig. 4 is an overview of a node facility consisting of 10 neurons within HNoC, this is interfaced directly with the astrocyte-NoC which is structured in a ring topology. The astrocyte communicates with all 10 neurons in this single facility.

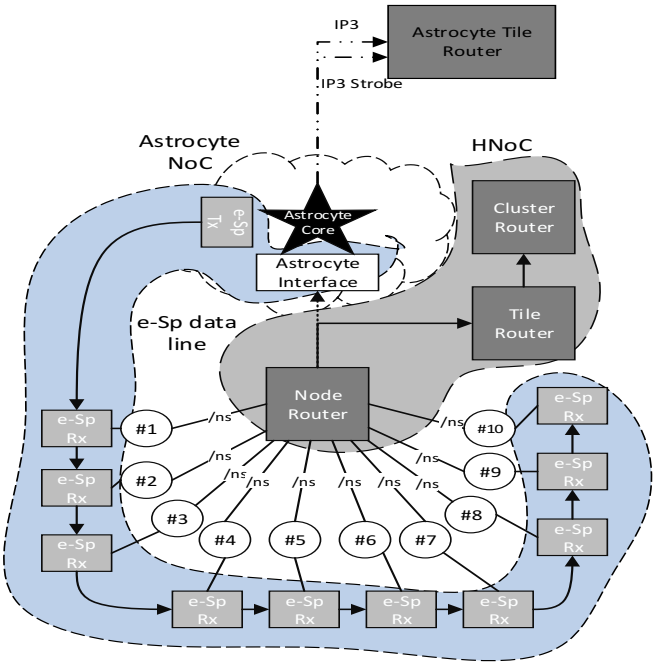

Fig. 4. HNoC interfaced with an astrocyte core and its ring topology.

The e-SP signal, released by the astrocyte, is a global signal and is communicated to every synapse (*ns* is the max number of synapses per neuron). Ns is the number of synapses per neuron based on HNoC [2], this is a variable which may accommodate more neurons if necessary. The variable is used to show we are not limited by the number of synapses, however, the number of synapses has been limited to 10 in all computational models (where using an astrocyte). The synapses of each individual neural cell within this node facility are interfaced with the astrocyte core using an 'e-SP comms' block. An e-SP packet is transmitted from the e-SP TX block and circulated serially through the e-SP Rx modules i.e. enables all 10 neurons to receive the e-SP data from the astrocyte core. The SNN communicates spike information in parallel with the astrocyte ring communication. The astrocyte core computes e-SP based on the rate of activity (spike events) with which the neurons are communicating to the core via the node router. These NoC networks are separate as the information communicated is different; spikes are simply events while e-SP is a numerical value. The main communication paths are outlined in Fig 5. The astrocyte cell is depicted as a star and the neurons as circles. This illustrates the key signals in the astrocyte communication process in relation to previous work [7].

4.1 Astrocyte-to-Astrocyte Interconnect

Astrocytes communicate with other astrocytes via a separate communication protocol. This communication within the neuro-glia network can be considered a multi-level communication infrastructure supporting both local and global communication exchanges. The astrocyte supports information exchanges between multiple astrocytes (global) and also neurons (local) within a neuro-glia network. The global astrocyte-to-astrocyte channel communicates IP_3 data. Within an astrocyte, IP_3 can be considered as a pool of water connected to a reservoir, this reservoir extends to neighbouring astrocytes and maintains similar level of IP_3. When the IP_3 level drops in one, the other reservoirs exchange IP_3 to provide an equal balance across all pools.

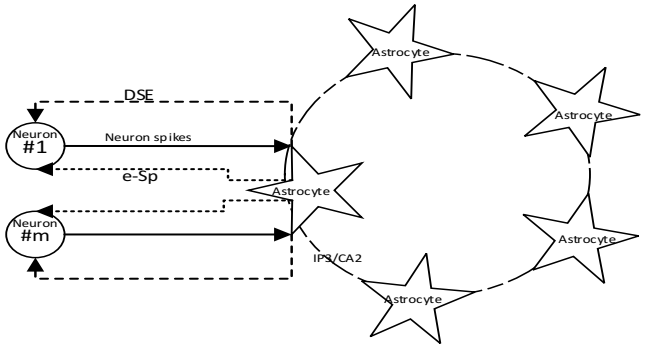

Fig. 5. Communication signals within a neuro-glia network.

4.2 *Global communication*

The astrocyte receives spike stimulus from the SNN, were event data is communicated from HNoC to the astrocyte via an additional output port within the node router of HNoC. The node router sends the data simultaneously to both the tile router of HNoC and the astrocyte core. The astrocyte model used, consists of two 2-AG generators and an astrocyte process; received spikes from neurons stimulate the release of 2-AG and the astrocyte produces the IP_3, DSE and e-SP signals. The IP_3 is a global communication signal within the astrocyte network, were it needs to be shared across neighbouring astrocytes. Each astrocyte has a pool of IP_3 and changes in this IP_3 indicate either increased or decreased levels. Astrocytes function by balancing and sharing their levels of IP_3 to ensure that there is enough IP_3 to facilitate repair and maintain normal functionality. Therefore, at an abstract level, the spike events from HNoC can

trigger changes in an astrocyte's IP_3 level, and this value must be communicated to other astrocytes (global communication). A multi-level infrastructure for communicating signals (both local and global) is shown in Fig. 4. This outlines the local communication of e-SP from the astrocyte, the e-SP Tx interfaces with the astrocyte core, and sends information down to each e-SP Rx. The global signal from each astrocyte is also connected to an astrocyte tile router, this is through the IP_3 signal, and a more detailed global communication is shown in Fig. 8. The astrocyte router has two main roles: (1) receive IP_3 level data from up to eight astrocytes and, (2) calculate the average IP_3 level for all eight astrocytes and communicate this back to all eight. The astrocyte core represents IP_3 as a 64-bit packet [26]. The rate at which IP_3 changes is much slower than spike events; typically 2-3 orders of magnitude slower. The key objective for hardware is to balance the physical area per astrocyte tile router facility while also meeting real-time requirements of the IP_3 exchange and update process. The astrocyte cluster facility is also an important component of the overall architecture of the astrocyte router.

The ring topology in NoCs has previously shown benefits in area-speed trade-offs for area for both SNN and neuro-glia hardware [36], [37], [28]. By exploiting the slower communication speeds of the biological IP_3 signal a time-multiplexed approach using ring structures can reduce area.

4.3 *Astrocyte Tile: Inter-router Module*

The 64 bit precision of the IP_3 data is significant and becomes area inefficient to communicate if done in traditional parallel-line channels. The astrocyte tile router is comprised of three main components, an adder, a ring interface and an update manager. Each astrocyte is attached to an inter router which manages the parallel to serial conversion and ring transmission interface. When an astrocyte has an updated IP_3 value it releases an IP_3-vld signal to the inter-router, this is followed by its 64-bit IP_3 value. The 'inter router' accepts this data value and consequently requests a token from the update manager, this will be explained in more detail within the next section. Due to the potential for numerous IP_3 values (changing within a short timeframe) from a number of astrocytes, it is important to manage the token requests and the communication process, this will allow the astrocytes to remain in sync. The inter router will request a token from the update manager, the update manager accepts the request, if the token is free, the update manager grants the token to the requesting 'inter router'. This process starts the chain of events regarding the IP_3. This is the main process regarding global communication in an astrocyte network. When an 'inter-router' token has been granted the 'inter router' will serially transmit its IP_3 data to the IP_3 accumulator via a PISO contained within each 'inter router' unit. This serial link enables a reduction in the physical wiring. The inter-router sends its value serially, and simultaneously transmits a signal to inform the next 'inter router' to subsequently send its data to the IP_3 accumulator. Each 'inter-router' sends its data serially whilst sending a start signal to the next 'inter-router'. This data propagates through the IP_3 accumulator, adding each astrocytes IP_3 value and sending back an average value of IP_3 to the astrocytes. The resulting new IP_3 value is serially propagated back through each 'inter router' facility using the ring topology. The IP_3 data is sent in a 64 bit bus to the 'inter router', and sent serially via a single wire. Use of the serial ring topology minimizes the wire overhead set by the large packet size and reduces the usage of buses as a single data line is used between each 'inter-router'. The 'inter-router' uses four separate data lines to communicate back and forth with the astrocyte. The 'inter-router' then communicates to the update manager and IP_3 accumulator. It then has five wires, the first wire is for receiving a token from the update manager and the remaining four are used to manage the IP_3 within the ring topology. Overall, the astrocytes send data to the astrocyte tile router and the IP_3 accumulator manages the incoming data from all astrocytes via a multiplexer, Fig.6 illustrates the packet layout, each IP_3 value is in a 64-bit packet, astrocyte #0 contains one start bit, making the packet 65-bits long, and the remaining astrocytes use 64-bit packets. Each 64-bit packet is numbered from 0 to 63, as shown below. Fig. 7 shows the IP_3 accumulator in more detail as it manages and accumulates 8x64 bit serial values. The output of multiplexer is connected to a serial adder circuit. The adder accumulates all eight IP_3 values and forwards this to the 'Divider' where a shift-by-3 operation is performed to complete the IP_3 averaging process. The 'Ring I/F' interface uses the ring of the inter-routers to communicate the average IP_3 value back to each astrocyte. Fig. 8 highlights all the input and outputs in regard to each 'inter-router'. The 'inter-router' manages the values from the astrocytes and keeps the astrocyte tile router working in a very specific manner, accumulating and averaging IP_3 values and communicating this information back to the astrocytes within the network.

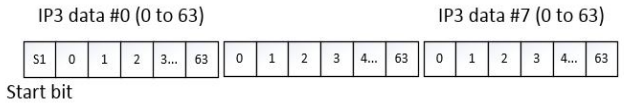

Fig. 6. Packet layout in closer detail.

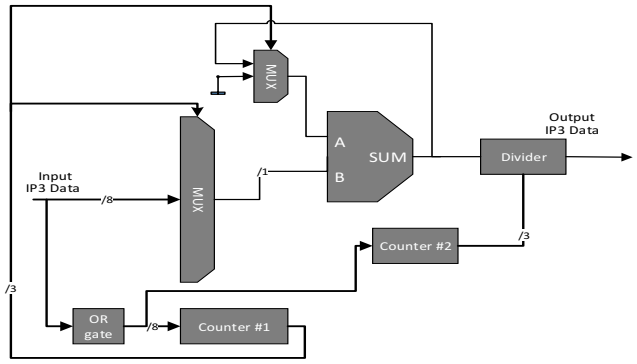

Fig. 7. IP_3 accumulator in closer detail.

4.4 *Update Manager*

In this work, groups of eight astrocytes were selected to provide an interconnect architecture with an optimized astrocyte to tile router (astrocyte) model. This allows an 8:1 ratio of astrocytes to tile router (astrocyte), based on a biologically realistic model, as astrocytes have approximately 6/8 neighbours [38]. This architecture allows each astrocyte to communicate between 7 neighbouring astrocytes, this is true for each astrocyte tile router, allowing an efficient and biologically inspired model.

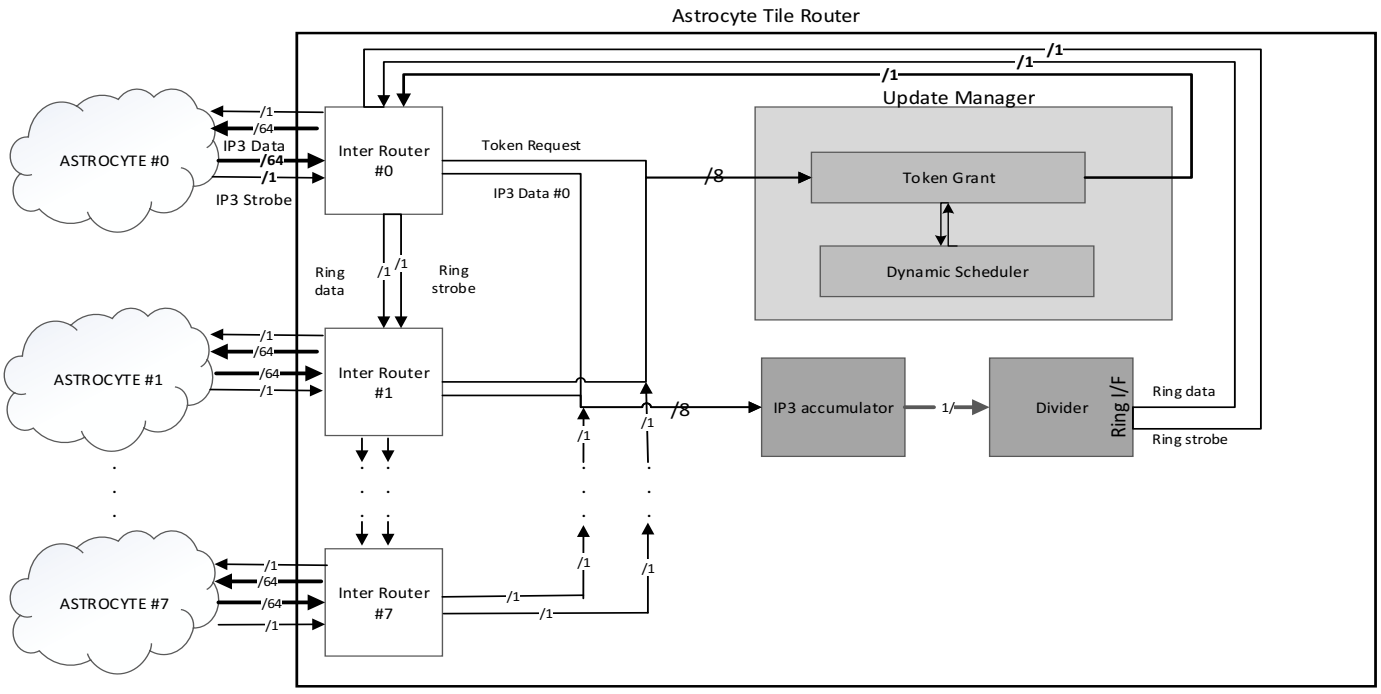

Fig. 8. Astrocyte tile router communications. IP_3 from the astrocyte is sent to the inter router. The inter router requests a token. If the token is granted, each astrocyte sends its IP_3 value to the IP_3 accumulator. The updated IP_3 value is then sent back serially via the ring interface to each astrocyte.

The importance of the update manager is twofold, managing the synchronisation of the input IP_3 data from the astrocytes and 'inter-routers' into the multiplexer. Secondly, it enables the control of the rate at which the update or averaging process is done. IP_3 values change very slowly. The update manager holds a single token, and when an 'inter-router' requests this token, the update manager checks the status of the token. If it is free, the token is granted to 'inter-router #0'. If it is not free, the request is ignored. This process allows scope for a more efficient and fair token system. Therefore, using a dynamic scheduler to manage the token using variables, it provides a more efficient and effective strategy. The values of the IP_3 will change constantly, resulting in more frequent IP_3 outputs. The change of IP_3 will be insignificant at times, and therefore, the rate of change from the astrocytes may be reduced. The overall IP_3 shared in the network will not change dramatically. To perform a complete update with every change from every astrocyte would be inefficient. Therefore, rather than perform an average IP_3 calculation each time a single astrocyte requests the action, the dynamic scheduler provides a means by which to minimise unnecessary averaging calculations. This reduces power consumption as two thresholds are put in place. The variables used are the number of token requests from the astrocytes and a max time period between token requests. Subsequently, when one of these thresholds are met, a computation or update will be performed.

4.5 *Dynamic Scheduler*

The dynamic scheduler (DS) manages requests by astrocytes to perform the IP_3 average and update processes. As mentioned previously, it uses two variables to do so. The token is released when either A) a number of token requests from the 'inter-routers' has been requested (n_{toc}) or B) a max time period (T_{DS}) has passed. For example, to save power in the astrocyte tile router the DS only schedules an update process when one of two of the conditions are met: 1) when 3 or more token requests are received or 2) when at least one token request received and a max time period between token requests has been reached.

Fig. 9 is a flow chart detailing the DS process. The update manager waits for an initial token request. When a token request is received, the n_{toc} increments to 1 and a timer starts, T=0. This is the start of the update process. The system waits for another token and checks that T is under T_{DS} if the update time window has expired, without receiving another token request the update will start. This is due to the inherent slow biological timescale, T_{DS} is a variable threshold, and this threshold is started when the first token request has been sent, regardless of the number of subsequent token requests, this ensures that the system updates periodically with changes in IP_3. If there are three or more token requests within this time period (T_{DS}) the DS will enable an update of the IP_3 values by granting the token to 'inter-router #0'. Thus, the DS aims to reduce the frequency of the update based on two conditions. The key operations of the DS are depicted in Fig.9. The update window T_{DS} is based on timescales of 10ms, 100ms and 1 sec. These are the max update rates of IP_3, where the refresh/update rate is not immediate but rather dynamic within the time constraints of update window. If there are more than 3 token requests, N_{toc}, within this time frame the IP_3 will update. If there is one token request the update will not occur until the update window expires. More experiments will be carried out to optimise and explore these threshold variables as there is an important balance between performance and efficiency.

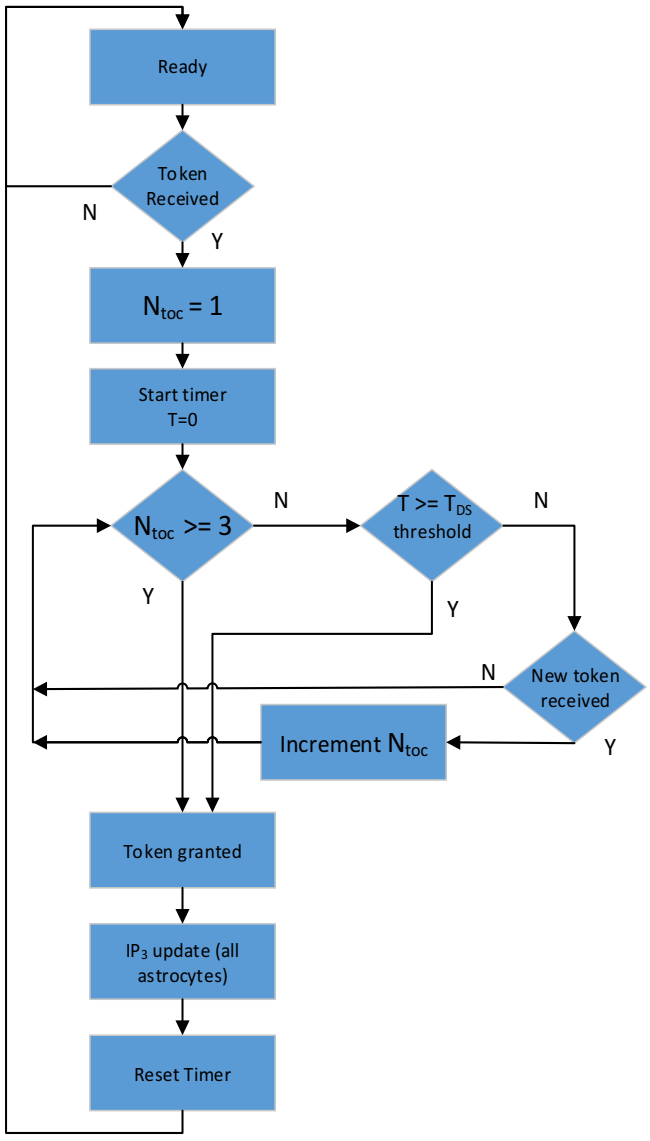

Fig. 9. Update manager DS flow chart.

5. Results and Analysis

This section outlines the testbench and provides area-power performance analysis of the astrocyte tile router. The performance of the astrocyte tile router's NoC ring topology is compared with the astrocyte core itself (computation component) and the spike-based HNoC (interconnect) to demonstrate its compactness and hardware scalability.

The HNoC neuron facility, astrocyte cell and proposed astrocyte router have been described in VHDL and synthesised for a Xilinx Virtex-7 XC7VX485T-2FFG1761C FPGA evaluation board using Xilinx Vivado 2016.4.

A. Performance: Astrocyte tile router

The area and power estimates are obtained using Vivado, which estimates the area to be utilised on board the FPGA and compares metrics using LUTs (look up tables) and slice registers. Power is compared using both the static and dynamic power, this evaluation allows comparisons of the astrocyte components. Table 1 (below) outlines the area overhead of the astrocyte tile router and compares it with the astrocyte core, HNoC neuron facility and "e-SP comms". Therefore, the 64-bit data to be communicated, is broken down serially and communicated using a slower ring-topology with serial communication to minimise area. The astrocyte tile router maintains the 64-bit resolution capacity of the computational model.

In Table 1, the astrocyte core (i.e. computation component) is used as a benchmark to compare area size of the various interconnect components. The HNoC neuron facility is around 3.2% in terms of LUTs and 4.5% in terms of slice registers. The e-SP comms interconnect is very compact with only 0.6% of LUTs and 1.2% in slice registers. The astrocyte tile router is 2.2% in terms of LUTS and 4% in terms of slice registers which is also compact; smaller interconnect area requirements than the Node Router (HNoC).

Table 1. Astrocyte Component comparison

Component	LUTs	(%)	Slice Register	(%)
Astrocyte Core	16,305	-	16,182	-
Node Router (HNoC)	527	**3.2**	735	**4.5**
AstrocyteTileRouter	365	**2.2**	651	**4**
e-SP comms	99	**0.6**	199	**1.2**

Note: there is a small area overhead incurred by the astrocyte tile router and the 8 inter-routers. These inter-routers act as signal managers, directing the IP_3 from the astrocyte core to the astrocyte tile router, each tile consumes 70 LUTS and 64 slice registers, and the inter-router uses 27 LUTs and 55 slice registers. However, this design must be analysed as a whole entity and therefore, will be referred to as such.

B. Scalability Analysis

The scalability, refers to how this design will scale as the size of the network scales. Fig.10 shows the interconnect area overhead for various sizes of neuro-glia network implementations, as the number of neurons increase and the size of the network scales, the astrocyte communication must scale accordingly; e.g. in a network with increasing node facilities of 10x10 up to 50x50, there must be an according number of astrocytes and tile routers (astrocyte). Fig.10 shows that the astrocyte tile router interconnect scales proportionately as the size of the network scales. However, the number of LUTs and slice registers increase exponentially as the number of tiles increases. Comparing one neuron facility of HNoC and one instance of the 'e-SP comms' block previously mentioned, it allows the scale and impact of the tile router (astrocyte) on the network to be compared this shows a nominal overhead. As there are ten neuron facilities per tile facilities in HNoC and each tile facility correlates with a 1:1 ratio of tile facilities (astrocyte). The secondary axis compares area using Slice registers and this is scaled up from 1x1 to 50x50, these are represented by the dashed lines.

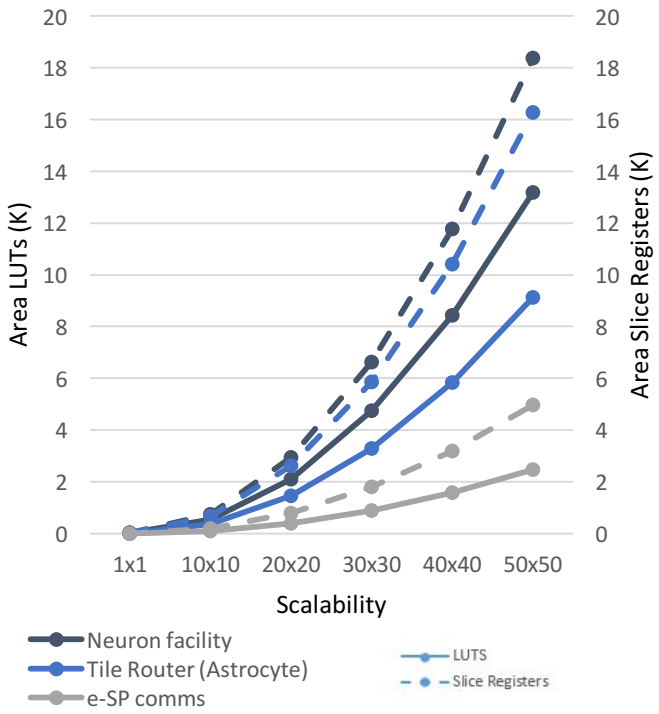

Fig. 10. Scalability LUTs and slice registers.

These results indicate that there is a very small area overhead when communicating IP_3 between astrocytes and this is important for global self-repair. The astrocyte core uses 64-bit precision and as a result this affects the area overhead of the astrocyte tile router design. The computational model uses double point precision and the astrocyte requires this precision As it is required to remain accurate with the computational requirements. The results show that the area overhead incurred by adding the astrocyte tile router and 'e-SP comms' interconnect block, this is small when compared to the size of the actual astrocyte computation core. There is one astrocyte tile router to 8 astrocytes and each astrocyte has an "e-SP comms" block, this shows the scale of the interconnection to computation, 1 astrocyte tile: 8 astrocytes: 8 "e-SP comms" blocks.

C. Power Analysis with Dynamic Scheduler

Table 2 shows that as the update period, t_{DS}, and as this timescale increases, the power consumed reduces. As the astrocyte is typically a slow changing process in the order of seconds, this shows that there is scope to reduce the update rate in hardware. This will also reduce power to support scalable implementations. Table 2, compares the power consumed over the course of 1s. t_{DS} is the time interval of each update whilst updating the global IP_3 once within one tile facility (astrocyte). This is the power consumed solely by the tile facility. Therefore, 1 update is 2.456 Watts, if we update 10 times during this time period it is 24.56 Watts. Running at 100MHz, the tile router (astrocyte) requires 194 clock cycles per update, therefore, per read/average/update iteration for the 8 astrocytes. Therefore, each iteration consumes $4.7*10^{-6}$ joules per update. The energy consumed for 10 and 100 updates/sec can be extrapolated by multiplying 10 and 100 respectively. One entire update cycle, consists of adding the IP_3 values from each astrocyte. This update process takes an average of the IP_3 and communicating this new value around each astrocyte. Fig. 11. Shows that as we reduce the number of updates or iterations for a certain time period, it significantly reduces the power consumed, a slower update against the power consumed is variable to find the best performance metric.

Table 2. Power consumed per update threshold.

t_{DS} (ms)	10	100	1,000
Power (watts)	245.6	24.56	2.456

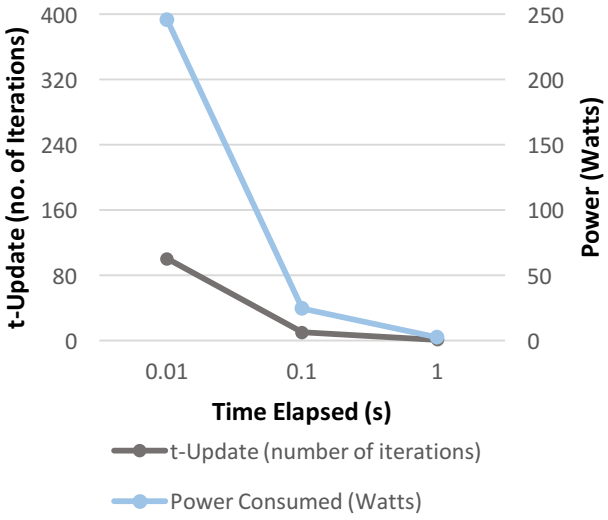

Fig. 11. Dynamic scheduler evaluation.

Furthermore, it is important to explore the power consumed during each update iteration and this was carried out using xPower within Vivado 2016.4. Using a typical activity file, Table 3 outlines the power consumed by the astrocyte tile router compared with the astrocyte computation itself, and the 'e-Sp comms' block. This demonstrates that the interconnect for the astrocyte communication is scalable as it only exhibits ~12% (3.071/25.471) of the power used by an astrocyte computation. Table 3 outlines the power from the overall design, the tile router and the inter-routers. This table compares using watts, and this is an average for one iteration and then scaled for 1, 10, 20, 30 and 40 iterations. Table 2 shows that the low power design consumes on average around 2.45 watts per iteration and it should be disclosed that this is broken down further into 1.28w (signals) and 1.24w (data). This design, shows a small area overhead, 2.2% in terms of LUTS and 4% in terms of slice registers relative to the astrocyte core. Fig.12. compares the power consumed as the interconnect scales, as the size of the network scales the power consumption scales linearly, this is preferred as the network is expected to be reproduced on large scales.

Table 3. Power Analysis.

	(Watts)		
Component	**Static**	**Dynamic**	**Total**
Astrocyte cell	0.732	24.739	**25.471**
Tile router (Astrocyte)	0.088	2.465	**2.543**
'e-SP comms' Block	0.074	0.454	**0.528**

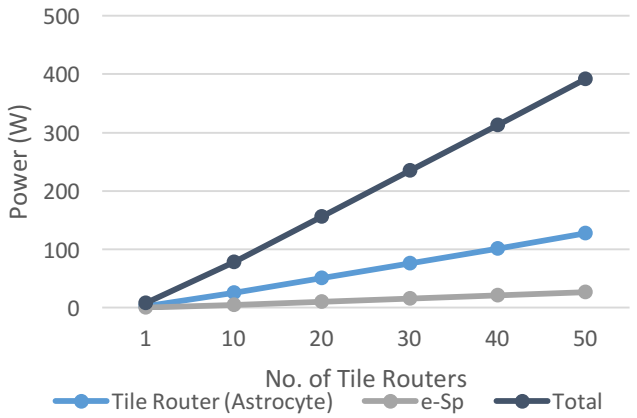

Fig.12. Scalability in terms of astrocyte tile routers.

Finally, the overall communication strategy addresses both local and global communication to enable neuro-glia network sizes which can be used to realize future self-repairing electronic systems.

6. Discussion and Conclusion

Developing a neuro-glia network in hardware requires an interconnect to consist of low overheads and a balance between speed and accuracy. As such it is important that both local and global communication mechanisms consist of low area and low power overheads. Astrocytes are computationally expensive and therefore, demand a lot of resources on an FPGA platform. With both individual networks, neural network and astrocyte network the number of processing elements (neurons and astrocytes) and communication signals is significant in size and interfacing the two networks is a difficult challenge. Previous work on high level astrocyte to astrocyte communications and combining local and global communication mechanisms, allows the astrocyte to use necessary resources whilst maintaining the low overheads in both area and power to provide a scalable solution to the interconnect challenge. The combination of using both the 'e-Sp' and the 'IP3' astrocyte router handshaking mechanisms for local and global communications within HNoC using NoC technology has provided a solution for a neuro-glia network which is in essence, a topology of two networks (astrocyte and neuron) interconnected. The use of a ring topology in the NoC provides a good trade-off between reducing area/wire overheads and relaxing the communication speed of data provided by the astrocyte to synapses/neurons, as astrocytes communicate at slow speeds in biological terms.

This novel NoC interconnection scheme for communicating enables a significant number of astrocytes to exchange data with other astrocytes with minimal area and low power constraints. Each astrocyte is interfaced with 10 neurons and each astrocyte tile router accommodates 8 astrocytes which allows 80 neurons per astrocyte tile facility. This in the future will enable self-repair within SNN hardware and helps us explore self-repair in electronics using biologically inspired systems. This proposed NoC interconnect provides a hardware building block for developing neuro-glia interconnect for self-repair strategies. Future work with neuro-glia networks aims to provide a distributed and fine grained self-repair using astrocytes in hardware based on the biological and computational models of previous works. In the future, this hardware will be used to explore self-repair leading to bio-inspired self-repair systems and applications. Based on HNoC, using a 3D concept on 2D hardware, of which the Tile router (astrocyte) is also modelled on. In the future we aim to further this work using a cluster facility (astrocyte). This will allow clusters of astrocytes to communicate total IP3 and Ca2+ which will allow the cluster to share information with other clusters to further realise the potential of this work. We aim to use the direct and indirect signals for strengthening and weakening of the synapses in the node router, we have a tile router and an astrocyte tile router and subsequently we aim to use a cluster facility to further scalability.

REFERENCES

[1] R. A. Shafik, J. Mathew, and D. K. Pradhan, "Introduction to Energy-Efficient Fault-Tolerant Systems," in *Energy-Efficient Fault-Tolerant Systems*, J. Mathew, R. A. Shafik, and D. K. Pradhan, Eds. New York, NY: Springer New York, 2014, pp. 1–10.

[2] S. Carrillo, J. Harkin, L. J. McDaid, F. Morgan, S. Pande, S. Cawley, and B. McGinley, "Scalable hierarchical network-on-chip architecture for spiking neural network hardware implementations," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 12, pp. 2451–2461, 2013.

[3] F. G. De Lima, Cota, L. Carro, M. Lubaszewski, R. Reis, R. Velazco, and S. Rezgui, "Designing a radiation hardened 8051-like micro-controller," *Proc. - 13th Symp. Integr. Circuits Syst. Des.*, pp. 255–260, 2000.

[4] M. Rebaudengo, L. Sterpone, M. Violante, C. Bolchini, A. Miele, and D. Sciuto, "Combined software and hardware techniques for the design of reliable IP processors," *Proc. - IEEE Int. Symp. Defect Fault Toler. VLSI Syst.*, pp. 265–273, Oct. 2006.

[5] S. Zhang and H. Liu, "Synthetical analysis on space radiation tolerance techniques in ASICs and FPGAs," *2011 Int. Conf. Syst. Sci. Eng. Des. Manuf. Informatiz. ICSEM 2011*, vol. 2, pp. 305–310, 2011.

[6] Y. Cai, Y. Zhao, and L. Lan, "Implementation of a reconfigurable computing system for space applications," *2011 Int. Conf. Syst. Sci. Eng. Des. Manuf. Informatiz. ICSEM 2011*, vol. 2, pp. 360–363, 2011.

[7] K. Kyriakoulakos and D. Pnevmatikatos, "A novel SRAM-based FPGA architecture for efficient TMR fault tolerance support," *FPL 09 19th Int. Conf. F. Program. Log. Appl.*, pp. 193–198, 2009.

[8] S. D'Angelo, C. Metra, S. Pastore, A. Pogutz, and G. R. Sechi, "Fault-tolerant voting mechanism and recovery scheme for TMR FPGA-based systems," *Defect Fault Toler. VLSI Syst. 1998. Proceedings., 1998 IEEE Int. Symp.*, pp. 233–240, 1998.

[9] S. Murali, T. Theocharides, N. Vijaykrishnan, M. J. Irwin, L. Benini, and G. De Micheli, "Analysis of error recovery schemes for networks on chips," *IEEE Des. Test Comput.*, vol. 22, no. 5, pp. 434–442, 2005.

[10] W. Barker, D. M. Halliday, Y. Thoma, E. Sanchez, G. Tempesti, and A. M. Tyrrell, "Fault tolerance using dynamic reconfiguration on the POEtic tissue," *IEEE Trans. Evol. Comput.*, vol. 11, no. 5, pp. 666–684, 2007.

[11] a. Alaghi, N. Karimi, M. Sedghi, and Z. Navabi, "Online NoC Switch Fault Detection and Diagnosis Using a High Level Fault Model," *22nd IEEE Int. Symp. Defect Fault-Tolerance VLSI Syst. (DFT 2007)*, pp. 21–29, 2007.

[12] K. Reick, P. N. Sanda, S. Swaney, J. W. Kellington, M. Mack, M. Floyd, and D. Henderson, "Fault-tolerant design of the IBM Power6 microprocessor," *IEEE Micro*, vol. 28, no. 2, pp. 30–38, 2008.

[13] S. Mitra, W.-J. Huang, N. R. Saxena, S.-Y. Yu, and E. J. Mccluskey, "Reconfigurable Architecture for Autonomous Self-Repair," *IEEE Des. Test Comput.*, vol. 21, no. 3, pp. 228–240, 2004.

[14] J. Liu, J. Harkin, Y. Li, and L. Maguire, "Online fault detection for Networks-on-Chip interconnect," *Proc. 2014 NASA/ESA Conf. Adapt. Hardw. Syst. AHS 2014*, pp. 31–38, 2014.

[15] E. Painkras, L. A. Plana, J. Garside, S. Temple, F. Galluppi, C. Patterson, D. R. Lester, A. D. Brown, and S. B. Furber, "SpiNNaker: A 1-W 18-core system-on-chip for massively-parallel neural network simulation," *IEEE J. Solid-State Circuits*, vol. 48, no. 8, pp. 1943–1953, 2013.

[16] M. De Pittà, N. Brunel, and A. Volterra, "Astrocytes: Orchestrating synaptic plasticity?," *Neuroscience*, vol. 323. pp. 43–61, 2016.

[17] M. Naeem, L. J. McDaid, J. Harkin, J. J. Wade, and J. Marsland, "On the Role of Astroglial Syncytia in Self-Repairing Spiking Neural Networks," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 26, no. 10, pp. 2370–2380, 2015.

[18] J. J. Wade, L. J. McDaid, J. Harkin, V. Crunelli, J. A. S. Kelso, and V. Beiu, "Exploring retrograde signaling via astrocytes as a mechanism for self repair," *Proc. Int. Jt. Conf. Neural Networks*, pp. 3149–3155, Jul. 2011.

[19] J. Wade, L. McDaid, J. Harkin, V. Crunelli, and S. Kelso, "Self-repair in a bidirectionally coupled astrocyte-neuron (AN) system based on retrograde signaling.," *Front. Comput. Neurosci.*, vol. 6, no. September, p. 76, Jan. 2012.

[20] J. Liu, J. Harkin, L. McDaid, D. M. Halliday, A. M. Tyrrell, and J. Timmis, "Self-repairing mobile robotic car using astrocyte-neuron networks," *2016 Int. Jt. Conf. Neural Networks*, pp. 1379–1386, 2016.

[21] Y. Irizarry-Valle and A. C. Parker, "Astrocyte on neuronal phase synchrony in CMOS," *Proc. - IEEE Int. Symp. Circuits Syst.*, pp. 261–264, 2014.

[22] Y. Irizarry-Valle and A. C. Parker, "An astrocyte neuromorphic circuit that influences neuronal phase synchrony," *IEEE Trans. Biomed. Circuits Syst.*, vol. 9, no. 2, pp. 175–187, 2015.

[23] B. A. Abed, A. Ismail, and N. A. Aziz, "Real time astrocyte in spiking neural network," *SAI Intell. Syst. Conf. 2015*, pp. 565–570, 2015.

[24] H. Soleimani, M. Bavandpour, A. Ahmadi, and D. Abbott, "Digital implementation of a biological astrocyte model and its application," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 26, no. 1, pp. 127–139, Jan. 2015.

[25] M. Hayati, M. Nouri, S. Haghiri, and D. Abbott, "A Digital Realization of Astrocyte and Neural Glial Interactions," *IEEE Trans. Biomed. Circuits Syst.*, vol. 10, no. 2, pp. 518–529, 2016.

[26] J. Liu, J. Harkin, L. Maguire, L. McDaid, J. Wade, and M. McElholm, "Self-repairing hardware with astrocyte-neuron networks," *Proc. - IEEE Int. Symp. Circuits Syst.*, vol. 2016–July, pp. 1350–1353, 2016.

[27] J. Liu, J. Harkin, L. P. Maguire, L. J. McDaid, J. J. Wade, and G. Martin, "Scalable Networks-on-Chip Interconnected Architecture for Astrocyte-Neuron Networks," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 63, no. 12, pp. 2290–2303, 2016.

[28] G. Martin, J. Harkin, L. J. McDaid, J. J. Wade, J. Liu, and F. Morgan, "Astrocyte to spiking neuron communication using Networks-on-Chip ring topology," in *2016 IEEE Symposium Series on Computational Intelligence, SSCI 2016*, 2016.

[29] L. Benini and G. De Micheli, "Networks on chips: A new SoC paradigm," *Computer (Long. Beach. Calif).*, vol. 35, no. 1, pp. 70–78, 2002.

[30] W. J. Dally and B. Towles, "Route packets, not wires: on-chip interconnection networks," *Proc. 38th Des. Autom. Conf.*, pp. 684–689, 2001.

[31] A. Hemani, A. Jantsch, S. Kumar, A. Postula, J. Öberg, M. Millberg, and D. Lindqvist, "Network on a Chip: An architecture for billion transistor era," *Proc. Norchip - 2000*, pp. 166–173, 2000.

[32] S. Carrillo, J. Harkin, L. McDaid, S. Pande, S. Cawley, and F. Morgan, "Adaptive routing strategies for large scale spiking neural network hardware implementations," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6791 LNCS, no. PART 1, pp. 77–84, 2011.

[33] B. V. Benjamin, P. Gao, E. McQuinn, S. Choudhary, A. R. Chandrasekaran, J. M. Bussat, R. Alvarez-Icaza, J. V. Arthur, P. A. Merolla, and K. Boahen, "Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations," *Proc. IEEE*, vol. 102, no. 5, pp. 699–716, May 2014.

[34] J. Schemmel, J. Fieres, and K. Meier, "Wafer-scale integration of analog neural networks," *Proc. Int. Jt. Conf. Neural Networks*, pp. 431–438, Jun. 2008.

[35] R. Emery, A. Yakovlev, and G. Chester, "Connection-centric network for spiking neural networks," *Proc. - 2009 3rd ACM/IEEE Int. Symp. Networks-on-Chip, NoCS 2009*, pp. 144–152, 2009.

[36] S. Pande, F. Morgan, G. Smit, T. Bruintjes, J. Rutgers, B. McGinley, S. Cawley, J. Harkin, and L. McDaid, "Fixed latency on-chip interconnect for hardware spiking neural network architectures," *Parallel Comput.*, vol. 39, no. 9, pp. 357–371, 2013.

[37] J. Liu, J. Harkin, L. McDaid, and G. Martin, "Hierarchical networks-on-chip interconnect for astrocyte-neuron network hardware," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, vol. 9886 LNCS, pp. 382–390.

[38] C. S. von Bartheld, J. Bahney, and S. Herculano-Houzel, "The search for true numbers of neurons and glial cells in the human brain: A review of 150 years of cell counting," *J. Comp. Neurol.*, vol. 524, no. 18, pp. 3865–3895, 2016.